

# WAVEFRONT Stiffness Matrix Resequencing Program Modifications for the 1108 Computer

R. Levy  
DSN Engineering Section

*WAVEFRONT is a preprocessor computer program that provides resequencing to expedite decomposition of the structural stiffness matrix within the NASTRAN analysis system. Modifications described here improve WAVEFRONT by reducing core requirements, speeding the execution, and extending the acceptable categories for input data.*

## I. Introduction

WAVEFRONT (Refs. 1,2) is a preprocessor computer program that provides a resequencing of the structural stiffness matrix nodes for more efficient processing during structural analysis. The objective of WAVEFRONT is to provide a reordered nodal sequence to reduce the maximum stiffness matrix wavefront during decomposition. Input to and output from the program are designed for compatibility with the NASTRAN structural analysis system. The present modifications of WAVEFRONT are designed to extend its present capabilities and to improve its execution efficiency. The present resequencing strategy, however, will remain almost unchanged.

resequencing preprocessors. Current versions of NASTRAN (release Level 15 and earlier) employ a mixed bandwidth/active column algorithm during decomposition of the stiffness matrix. BANDIT attempts to reduce the matrix bandwidth by means of the Cuthill-McKee algorithm (Ref. 4) independently of the numbers of active columns. WAVEFRONT ignores the bandwidth and operates to reduce the number of active columns, which then become equivalent to the wavefront. For applicability to future releases of the NASTRAN program (beginning with Level 16), which will employ a pure wavefront decomposition algorithm, BANDIT has been amended (Ref. 5) to include new options developed by Gibbs, Poole and Stockmeyer (Ref. 6).

## II. Resequencing for the NASTRAN Program

Heretofore both the BANDIT (Ref. 3) and the WAVEFRONT programs have been found to be effective

Recently a promising approach has been developed by Gibbs (Ref. 7) and will possibly be included in BANDIT. This is a hybrid method that incorporates both a new

algorithm to develop a starting point and an existing algorithm of King (Ref. 8) to continue the resequencing process beyond this starting point. Although developed independently, the underlying strategy for resequencing within WAVEFRONT is closely related to King's strategy. It is quite likely, however, that the relative effectiveness of these, or other methods that may yet appear, is problem-dependent. Nevertheless, it appears to the writer that BANDIT may be preferable for regular structures with clearly defined and relatively smooth connectivity grids, while WAVEFRONT may be best suited to highly irregular structures of complex connectivity. Confirmation of this assessment of course awaits the dissemination of the new NASTRAN program and testing on a variety of diverse problems.

### III. WAVEFRONT Modifications

The purpose of these modifications is to improve program execution and scope of application, rather than to provide algorithm changes that affect the results produced. These modifications take advantage of special features specifically available on the Univac 1108, Exec 8 computer. It could require substantial coding changes to incorporate equivalent features that may be available on other computers. The modifications are in three categories:

- (1) Reduction of core storage and adding the capability of handling larger sized problems than heretofore.
- (2) Improvement of execution speed.
- (3) Extension of the type of NASTRAN connection cards that can be interpreted and processed.

#### A. Core Storage Reduction

The previous version of the program required storage for six arrays equal in size to the maximum number of connectivity matrix nodes and five arrays equal to the maximum number of connection edges linking pairs of nodes. It was also necessary to anticipate the upper bounds of this storage at compilation time. The modified version requires four vectors of node size and one vector of connectivity edge size and will automatically adjust to this size without recompilation. As an example, a problem previously compiled for 1200 nodes and 8000 edges that required 47.2K words of array storage would now require 12.8K storage locations. In addition to array storage, about 17K words of storage are required for code, fixed-size data blocks, and the program control table, so that for the

example described the total core would now be about 30K, while formerly it would have been about 60K. If two 65K banks of core are available, a problem of about 10,000 nodes and 65,000 edges could be resequenced.

This storage reduction has been achieved by elimination or equivalencing of storage arrays that were found to be redundant, and by use of the Fortran V Field (FLD) function, which allows packing of independent information within partial (half word) locations.

#### B. Execution Speed

Several of the subroutines were recoded to improve execution speed. Additional sorting operations were included by means of a more efficient sorting subroutine, and binary search algorithms were coded in-line to expedite locating particular elements within arrays. The time required for processing the input data appears to be less than half of that formerly required. After input is processed, the time required to perform a cycle of matrix nodal resequencing has been reduced by 30 to 40 percent. The typical time required for a complete program execution tends to be slightly more than half of the former time.

#### C. NASTRAN Connection Card Processing

NASTRAN connection card types are identified by the first three characters of the mnemonic appearing in the first three card columns. Twelve mnemonic types can now be recognized, while previously only eight were possible. The following three-character identifiers are now accepted:

CBA	CQU	CTR
CHE	CRO	CTU
CON	CSH	CTW
CQD	CTE	CWE

Only the integer labels of the connected node labels are read on these connections. These node labels must be placed in the proper NASTRAN 8-column card field but need not be right- or left-adjusted within the field. Up to two independent connections can be supplied on the CROd and CTUbe cards, but if only one connection is supplied it must appear in the left-most set of fields. The CHE (hexahedron elements) cards, which include a continuation card to supply the labels for the seventh and eighth nodes, require that the continuation card follow immediately after the associated mnemonic card.

The connectivity edges developed from these connection cards consist of all combinations that can be formed

from the set of associated node labels. In the case of the hexahedron and wedge connections, the present form of the NASTRAN element stiffness matrix will not develop all of these edges, but this is expected to have a minor influence on the sequencing produced.

#### IV. Input Data Deck and Runstream

The form of the bulk data consisting of a set of NASTRAN GRID cards followed by the connection cards is similar to that in previous versions of WAVEFRONT. However, as a new alternative, an existing set of NASTRAN SEQGP cards can be inserted in place of the GRID cards. These will redefine the initial nodal sequencing for attempts at improvement. Otherwise the initial sequence is established by the submitted order of the grid cards. A restriction on the SEQGP cards requires that all integer data on these cards be right-adjusted in their 8-column card fields. A set of SEQGP cards produced by a prior sequencing run on WAVEFRONT will meet this requirement. A mixture of GRID and SEQGP cards can also be supplied, providing that they generate a unique set of all of the nodes that are referenced by the connection cards. User-option parameters to control the manner of execution are supplied by a new namelist form that replaces and extends the previous fixed format option card. The optional data on this namelist "INPUT" are:

**KCYCLE** The number of cycles of resequencing cycles to be performed. Default = 1.

**ROWA** Integer row number of the initial sequence at which the first resequencing cycle should start. If this is not supplied, the first cycle will begin at a row of minimum connectivity. For subsequent cycles, the first row will be picked at random from the set of rows that do not have more than the median number of connections.

**IREP** The program will terminate if IREP consecutive cycles do not produce a reduction of the maximum wavefront. Default = 8.

**IPCH** For a cycle that successfully reduces the current maximum wavefront, SEQGP cards will be punched only if the new maximum wavefront is less than IPCH. Default = 5000. To prevent ever punching SEQGP cards, supply IPCH = 0.

**NPRT** If any non-zero integer is supplied, a connectivity table of the input sequence is printed. This table gives the row connec-

tivities in terms of nodal sequence number and external label. Default = 0.

**NEWCNT** If any non-zero integer is supplied, a row-by-row wavefront table and summary statistics will be printed for any sequencing cycle that produces a wavefront reduction. The statistics are useful for estimating the time that will be required to decompose the stiffness matrix.

**MGRIDS** This is an upper bound estimate of the maximum number of nodes. The default value is 4000, and since this affects only one storage vector, there is little reason to supply a lower number if the problem is appreciably smaller. If the number of nodes is larger than the default value, an estimated upper limit must be supplied.

**MSHIFT** This controls the magnitude of the steps in core size to accommodate the actual problem. While the connection cards are being read in, core is increased in jumps of MSHIFT to accommodate these connections. The default value of 1536 has been proposed, as reasonable and moderate changes in this would have little effect on execution efficiency.

**IDEBUG** If any non-zero integer is supplied, a very large amount of programmer debugging information will be supplied. The default value of 0 should not be changed.

As an example, for a problem estimated to contain less than 4000 nodes, where four sequencing cycles are to be attempted and no SEQGP cards are desired unless the wavefront is less than 40 nodes, the following cards could be supplied:

```
b$INPUT   KCYCLE = 4,   IPCH = 40,
```

```
NEWCNT = 123456,   $END
```

The modified program is cataloged on the JPL 1108 A computer. A sample runstream is supplied below:

```
@RUN .....
```

```
@ASG,A 52219*RIL.
```

```
@HDG,P optional heading
```

```
@XQT 52219*RIL.WAVE/TRIAL
```

Title Card, Format (80A1) For problem identification and echo

b\$INPUT

..... optional input parameters as described

b\$END

Header Card, Format (12A6) This will be echoed

Package of GRID, SEQGP, or a combination of these cards

Blank Card (at least the first 16 card columns must be blank)

Header Card, Format (12A6) This will be echoed

Package of connection cards, containing any of the 12 3-word mnemonics in the first three columns or one continuation card following each CHE card

Blank Card (at least the first 3 card columns must be blank)

@FIN

## References

1. Levy, R. "Resequencing of the Structural Stiffness Matrix to Improve Computational Efficiency" in *Quarterly Technical Review*, Vol. I, No. 2, pp. 61-70, Jet Propulsion Laboratory, Pasadena, Calif., July 1971.
2. Levy, R. "Structural Stiffness Matrix Resequencing Program (WAVEFRONT)," *COSMIC Report*, NPO-13322, Computer Program Abstracts Vol. 5, No. 2, July 15, 1973.
3. Everstine, G. C., "The BANDIT Computer Program for the Reduction of Matrix Bandwidth for NASTRAN," *Computation and Mathematics Department Research and Development Report 3827*, Naval Ship Research and Development Center, Bethesda, Md., March 1972.
4. Cuthill, E., and McKee, S., "Reducing the Bandwidth of Sparse Symetric Matrices," *Applied Mathematics Laboratory Technical Note*, AML-40-69, Naval Ship Research and Development Center, Washington, D.C., June 1969.
5. Everstine, G. C., "Recent Improvements to BANDIT," *NASTRAN: User's Experiences*, NASA TM X-3278, pp. 511-521, Sept. 1975.
6. Gibbs, N. E., Poole, W. G., Jr., and Stockmeyer, P. K., "An Algorithm for Reducing the Bandwidth and Profile of a Sparse Matrix," *Institute for Computer Applications in Science and Engineering (ICASE) Report*, Hampton, Virginia, July 1974.
7. Gibbs, N. E., "A Hybrid profile Reduction Algorithm," *Technical Report II*, Mathematics Department, College of William and Mary, Williamsburg, Va., Aug. 1975.
8. King, I. R., "An Automatic Reordering Scheme for Simultaneous Equations Derived from Network Systems," *Int. J. Numer. Meth. Eng.*, Vol. 2, pp. 523-533, 1970.